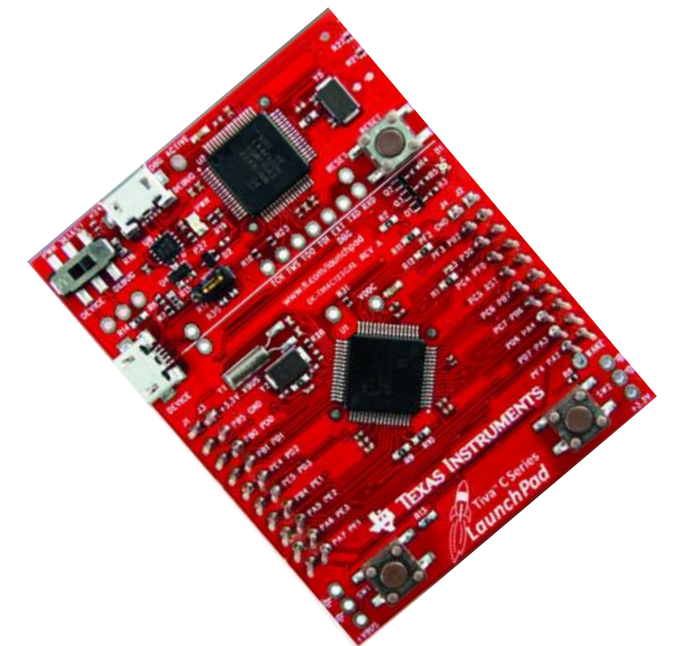
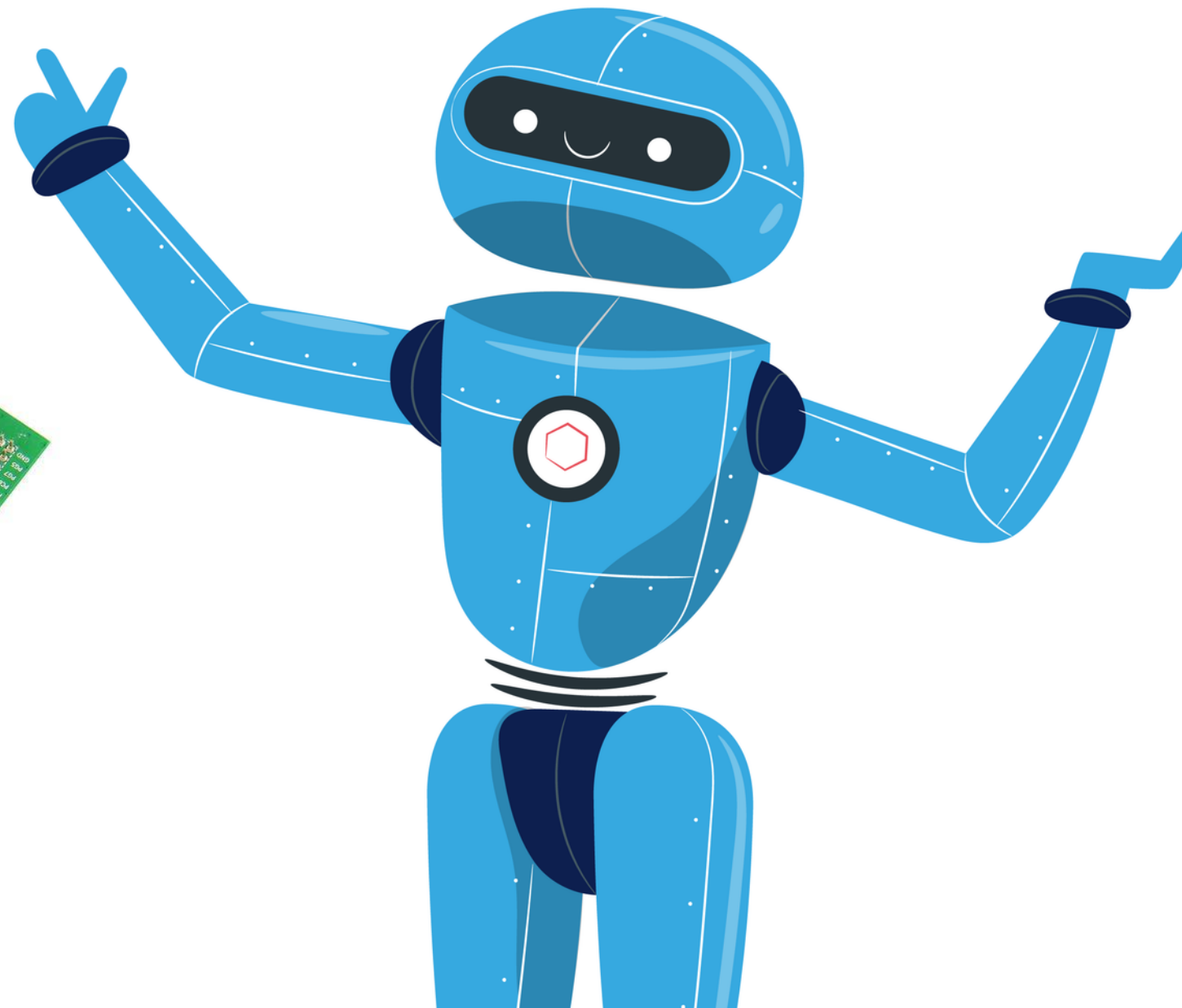
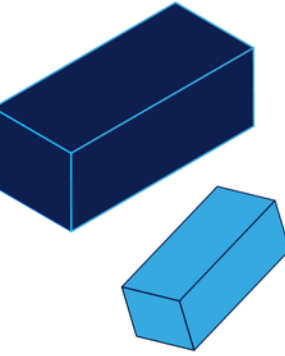


CLASE 9: ADC

MICROCONTROLADORES ARM

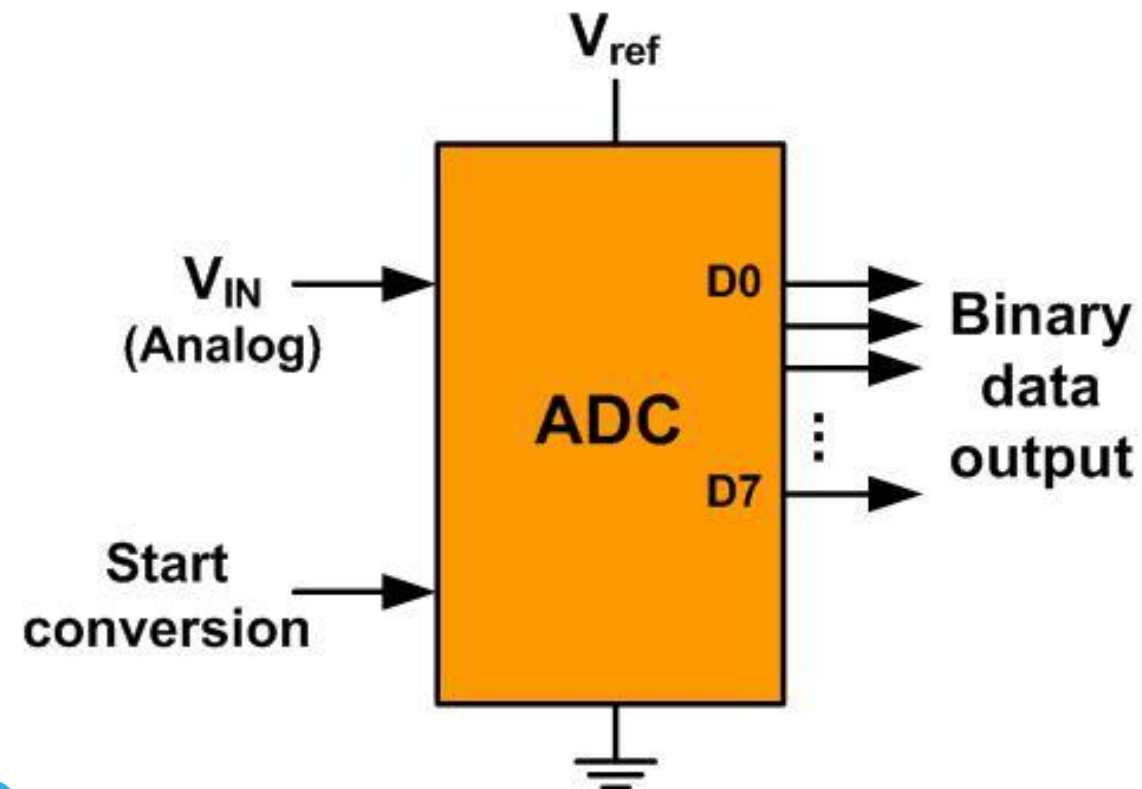




ADC



ADC



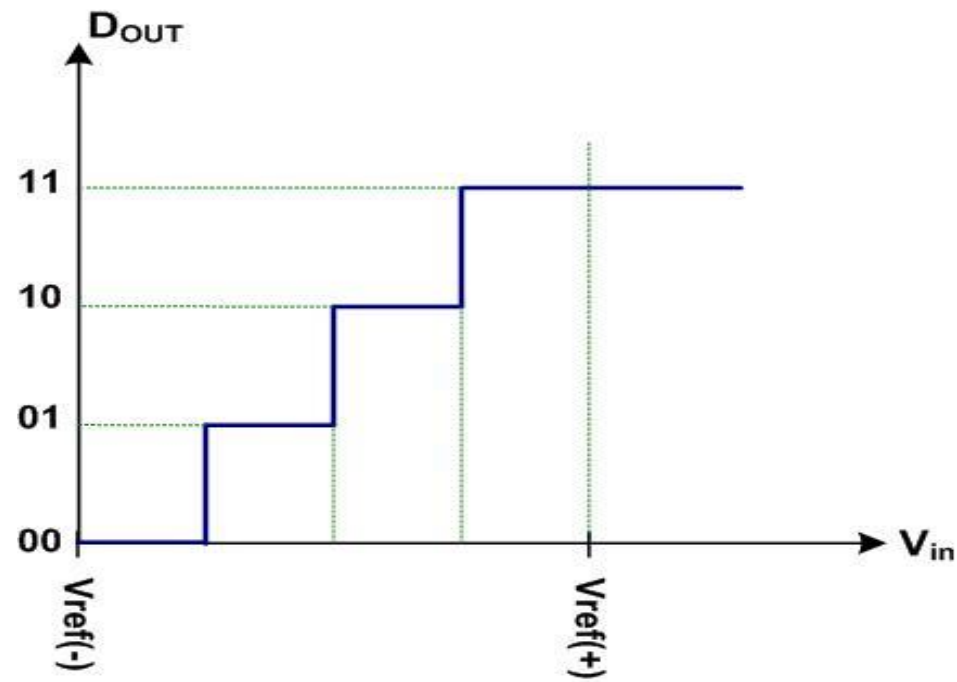
n-bit	Number of steps	Step size
8	256	$5V / 256 = 19.53 \text{ mV}$
10	1024	$5V / 1024 = 4.88 \text{ mV}$
12	4096	$5V / 4096 = 1.2 \text{ mV}$
16	65,536	$5V / 65,536 = 0.076 \text{ mV}$
Note: $V_{ref} = 5V$		

arm

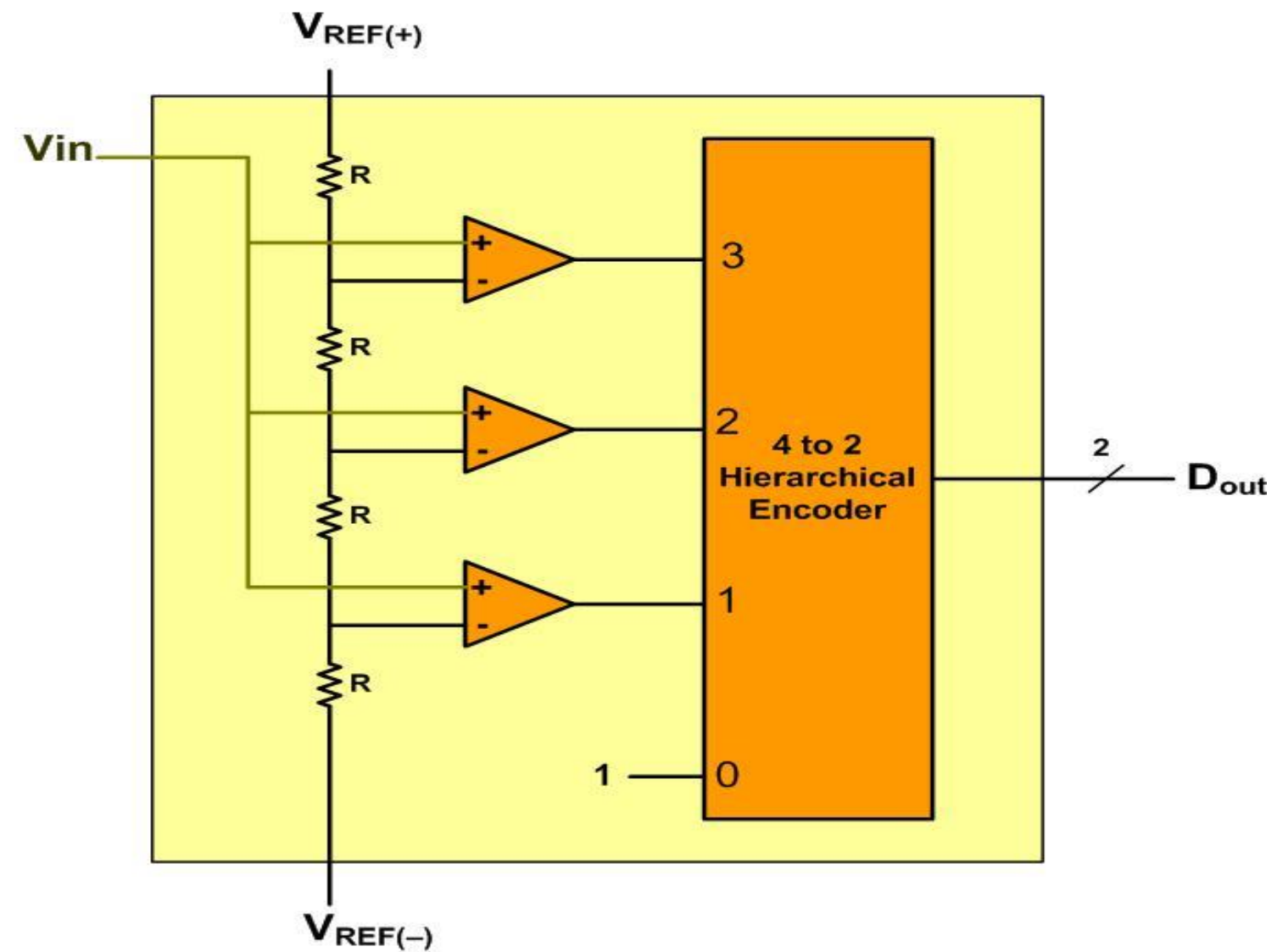
MICRO-
CONTRO-
LADORES
ARM

ADC

A Simultaneous 2-bit ADC



(a) The Relationship between V_{in} and D_{out}



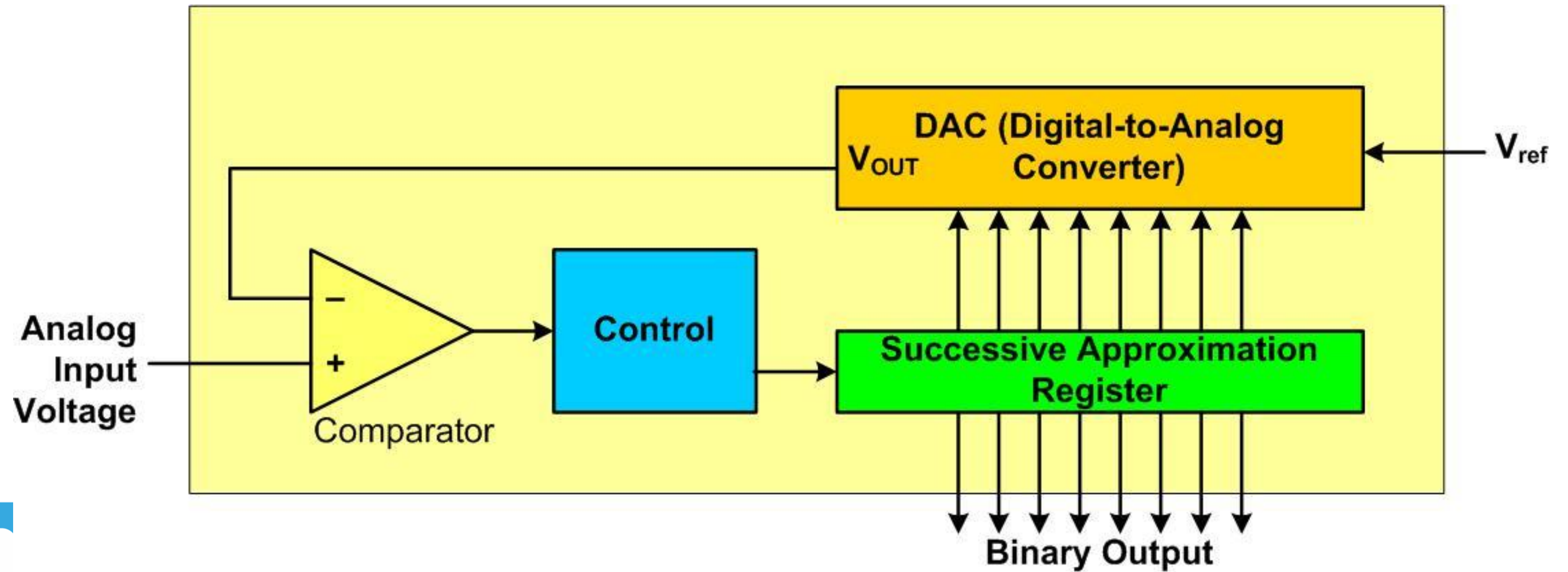
(b) The internal block diagram of a simple 2-bit ADC

arm

MICRO-
CONTRO-
LADORES
ARM

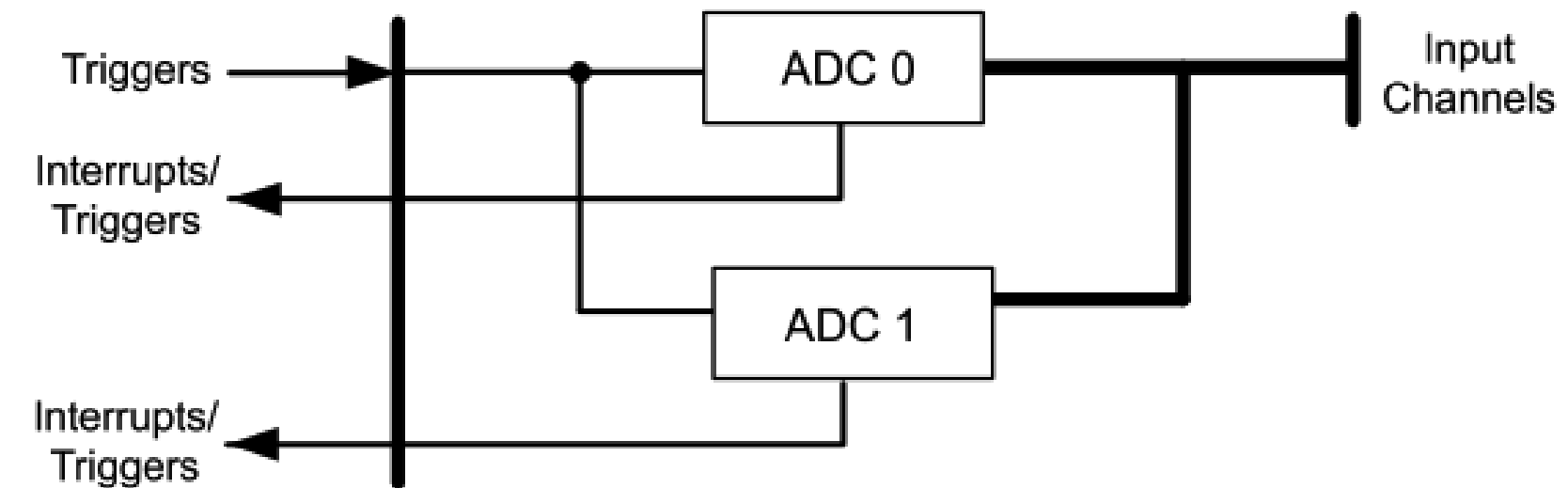
ADC

Successive Approximation ADC



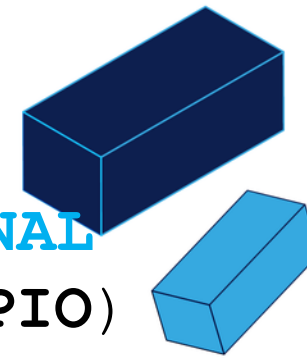
arm

MICRO-
CONTRO-
LADORES
ARM



ADC

CANALES ANALOGICOS



ADC Pin	GPIO Pin	Pin Type	Pin Function
AIN0	PE3	Input	ADC Analog Input Channel 0
AIN1	PE2	Input	ADC Analog Input Channel 1
AIN2	PE1	Input	ADC Analog Input Channel 2
AIN3	PE0	Input	ADC Analog Input Channel 3
AIN4	PD3	Input	ADC Analog Input Channel 4
AIN5	PD2	Input	ADC Analog Input Channel 5
AIN6	PD1	Input	ADC Analog Input Channel 6
AIN7	PD0	Input	ADC Analog Input Channel 7
AIN8	PE5	Input	ADC Analog Input Channel 8
AIN9	PE4	Input	ADC Analog Input Channel 9
AIN10	PB4	Input	ADC Analog Input Channel 10
AIN11	PB5	Input	ADC Analog Input Channel 11

PASOS PARA CONFIGURAR UN CANAL

- 1.Habilitar el reloj (**RCGCGPIO**)
- 2.Establecer el bit correspondiente en (**GPIOAFSEL**)
- 3.Deshabilitar la función digital (**GPIODEN**)
- 4.Establecer el bit correspondiente en (**GPIOAMSEL**)



ADC

Controles del secuenciador de muestra

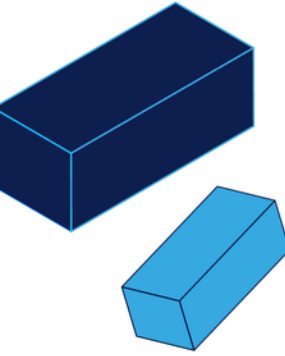
El control de muestreo y la captura de datos son manejados por los secuenciadores de muestras. Todos los secuenciadores son idénticos en implementación excepto por el número de muestras que se pueden capturar y longitud de sus FIFO. Cada muestra que se captura se almacena en el FIFO.

Sequencer	Number of Samples	Depth of FIFO
SS3	1	1
SS2	4	4
SS1	4	4
SS0	8	8

- Cada muestra se define en el registro **ADCSSMUXn** y el control de secuencia de muestra con el registro **ADCSSCTLn**, donde **n** corresponde al numero de secuencia. El muestreo se inicia estableciendo los bits del registro **ADCPSSI**.
- Una vez finalizado el muestreo y la conversión, los resultados son almacenados en los registros FIFO.

arm

MICRO-
CONTRO-
LADORES
ARM



EJEMPLO

arm

MICRO-
CONTRO-
LADORES
ARM

ADC

EJEMPLO CONFIGURACION

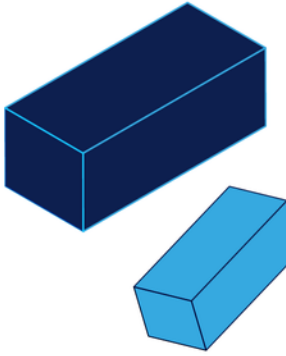
```
void ADC0_Config(void){
    /**
     * GPIO PORT CONFIG
     */
    /**1. enable clock*/
    SYSCTL_RCGCGPIO_R |= SYSCTL_RCGCGPIO_R4;
    while(!(SYSCTL_PRGPIO_R & SYSCTL_PRGPIO_R4)); //GPIOE is ready
    SYSCTL_RCGCADC_R |= SYSCTL_RCGCADC_R0;
    while(!(SYSCTL_PRADC_R & SYSCTL_PRADC_R0)); //ADC0 is ready
    /**2. Enable AFSEL bits*/
    GPIO_PORTE_AFSEL_R |= AN1 | AN2 | AN8;
    //GPIOE->AFSEL |= AN1 | AN2 | AN8;
    /**3. Clear DEN bit*/
    GPIO_PORTE_DEN_R &=~ (AN1 | AN2 | AN8);
    /**4. Disable the analog isolation circuit */
    GPIO_PORTE_AMSEL_R |= AN1 | AN2 | AN8;
    /**
     * Module Initialization
     */
    /**1. source CLOCK of ADC*/

    /**2. application, reconfigure the sample sequencer priorities */
    ADC0_SSPRI_R &=~(0x3U<<4);
    ADC0_PC_R = 0x01;
    /**
     * Sample Sequencer Configuration
     */
    /**1. Disable ASENn bits on ACTSS register*/
    ADC0_ACTSS_R &=~ (ADC_ACTSS_ASEN1 );
    /**2. Configure the trigger event for the sample sequencer in the ADCEMUX register.*/
    ADC0_EMUX_R &=~(0xF<<4);
    /**3. When using a PWM generator as the trigger source, use the ADC Trigger Source
    (ADCTSSEL) register to specify in which PWM module the generator is located.*/

    /**4. For each sample in the sample sequence, configure the corresponding input source in the
    ADCSSMUXn register.*/
    ADC0_SSMUX1_R |= 0x8<<8 | 0x2<<4 | 0x1; //1st-> AN1, 2nd -> AN2 and 3rd->AN8
    /**5. For each sample in the sample sequence, configure the sample control bits */
    ADC0_SSCTL1_R |= 1<<11 | 1<<10 | 1<<9; //TS2 selected, interrupt enable and 3rd sample end
    /**6. If interrupts are to be used, set the corresponding MASK bit in the ADCIM register.*/
    /**7. Enable the sample sequencer logic by setting the corresponding ASENn bit in the ADCACTSS register.*/
    ADC0_ACTSS_R |= ADC_ACTSS_ASEN1;
```

```
#define AN1
#define AN2
#define AN8
```

```
(1U<<2)
(1U<<1)
(1U<<5)
```



ar

MIC
CON
LAD
AF }

ADC

EJEMPLO CONFIGURACION

```
void ADC0_Read(void){
    ADC0_PSSI_R |= ADC_PSSI_SS1;
    while(!(ADC0_RIS_R & ADC_RIS_INR1));
    readVolt = ADC0_SSFIF01_R;
    readLm35 = ADC0_SSFIF01_R;
    readTemp = ADC0_SSFIF01_R;
    ADC0_ISC_R |= ADC_ISC_IN1;
}
```

```
int main(void){
    SYSTCL_RCGCGPIO_R |= 1<<5;
    GPIO_PORTF_DIR_R |= 1<<1;
    GPIO_PORTF_DEN_R |= 1<<1;
    ADC0_Config();
    for(;;){
        ADC0_Read();
        GPIO_PORTF_DATA_R ^= 1<<1;
        delay_ms(50);
    }
}
```

arm

MICRO-
CONTRO-
LADORES
ARM

UMAKER | CENTRO DE CAPACITACIÓN
DE DESARROLLO TECNOLÓGICO